CERTIFICATE OF MAILING OR TRANSMISSION 37 C.F.R. § 1.8
Pursuant to 37 C.F.R. § 1.8, I hereby certify that I have information and a reasonable basis for belief that this paper will be filed with the United States Patent and Trademark Office via the USPTO electronic filing system (EFS) on October 3, 2007.
Signature: R Zh — Date: 10-3-07

PATENT
016295.0624

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	§	
	§	
Nguyen et al.	§	
	§	
Serial No.: 09/768,665	§	Group No.: 2111
	§	
Filed: January 24, 2001	§	Examiner: Khanh Dang
	§	
For: System and Method for the Handling of System Management Interrupts in a Multiprocessor Computer System	§	

APPEAL BRIEF

Pursuant to 37 CFR § 41.37, a Notice of Appeal filed May 29, 2007, and a Notice of Panel Decision from Pre-Appeal Brief Review mailed July 3, 2007, Appellants hereby submit this Appeal Brief.

Petition for Extension of Time

Appellants petition for a two-month extension of time under 37 C.F.R. § 1.136 up to and including October 3, 2007. Appellants hereby authorize and instruct the U.S. Patent and Trademark Office to charge Deposit Account No. 02-0383 (matter 016295.0624) of Baker Botts L.L.P. in the amount of $970.00 for the fee associated with the extension of time ($460.00) and the fee for the filing of the appeal brief ($510.00). Appellants hereby authorize and instruct the

U.S. Patent and Trademark Office to charge Deposit Account No. 02-0383 (matter 016295.0624) of Baker Botts L.L.P. for any additional charges necessary for the filing of this response.

I. REAL PARTY IN INTEREST

The real party in interest is:

Dell Products, LP
One Dell Way
Round Rock, TX 78682-2244

II. RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision regarding this appeal.

III. STATUS OF CLAIMS

Claim	Status
1, 4-8, 16, and 19-23	Rejected
2-3, 9-15, 17-18, and 24-27	Cancelled
1, 4-8, 16, and 19-23	Appealed

IV. STATUS OF AMENDMENTS

No claim amendments have been filed subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In the following paragraphs, references to page and lines of the specification as filed are indicated in square brackets, *e.g.*, Page 1 line 28 through Page 2 line 12 would be indicated as [1:28-2:12] and Page 1, lines 28-32 would be indicated as [1:28-32]. Reference numerals and figure numbers are enclosed in parentheses.

Independent Claim 1 is directed to a method for handling system management interrupts (40-54) (FIG. 2) in a computer system (10) (FIG. 1) having multiple processors (12a-12d) (FIG. 1). Each of these multiple processors (12a-12d) is able to process a system management interrupt, but none of these processors (12a-12d) are dedicated to processing the system management interrupts of the computer system (10). [4:20-23] The method (FIG. 2) includes the following steps (40-54). [7:20-8:20] A predetermined signature (88) (FIG. 3) is written (42) to a predetermined register of a first processor (e.g. 12b). A software application command is executed (44) in the first processor (12b), causing the first processor (12b) to initiate (44) a system management interrupt. Each of the processors (12a-12d) receives (46) an instruction that a system management interrupt has been issued (46). Then, each of the processors (12a-12d) enters (48) system management mode. The content of each processor's (12a-12d) registers is saved (50) to each processor's (12a-12d) respective memory space (80-86) (FIG. 3). A second processor (*e.g.* 12c) is selected from the multiple processors (12a-12d) as a system management interrupt handler according to an arbitration scheme. The contents of the memory space associated with each processor (80-86) are scanned (52). When the second processor (12c) locates the predetermined signature (88) in one of the memory spaces (*e.g.* 82), the contents of the associated memory space (82) are used to obtain (54) any parameters necessary for the handling of the system management interrupt.

Independent Claim 16 is directed to a method for handling system management interrupts (40-54) in a computer system (10) having multiple processors (12a-12d). Each of these multiple processors (12a-12d) is able to handle a software system management interrupt, but none of these processors (12a-12d) are dedicated to processing the system management interrupts of the computer system (10). [4:20-23] The method includes the following steps (40-54). [7:20-8:20] A first processor (*e.g.* 12b) issues (44) an instruction to a chip set of the computer system (10). The chip set receives (46) the instruction, and, in response, issues (46) a command causing the processors (12a-12d) to enter system management mode (48). A software system management interrupt signature (88) is written (42) to a predetermined register of the first processor (12b) to indicate that the first processor (12b) issued (44) the command that caused the processors (12a-12d) to enter system management mode (48). The contents of each processor's (12a-12d) registers is written (50) to a memory location (78) including a memory space (80-86) reserved for and associated with the register contents of each processor (12a-12d). A second processor (*e.g.* 12c) is selected as the system management interrupt handler according to an arbitration scheme. The second processor (12c) includes a system management interrupt handler. A software system management interrupt is transmitted to the second processor (12c). The second processor (12c), in response to receiving the software system management interrupt, locates the software system management interrupt signature (88) in the memory location (78). The contents of the associated memory space (82) (the register contents saved (50) by the first processor (12b)) are retrieved (54) as parameters for use by the system management interrupt handler.

Independent Claim 22 is directed to a method for handling a system management interrupt (40-54) in a computer system (10) having multiple processors (12a-12d). Each of these multiple processors (12a-12d) is capable of acting as a software system management interrupt

handler, but none of these processors (12a-12d) are dedicated to processing the system management interrupts of the computer system (10). [4:20-23] The method includes the following steps (40-54). [7:20-8:20] Each of the processors (12a-12d) receives an instruction to enter (48) a system mode associated with the issuance (46) of a system management interrupt. A designated processor (*e.g.* 12c) is selected from the multiple processors (12a-12d) as a system management interrupt handler according to an arbitration scheme. The contents of a memory location (78) containing saved (50) contents of each processor (12a-12d) are scanned (52). A signature (88) identifying the saved (50) contents of the processor (*e.g.* 12b) that issued (44) an instruction causing the system management interrupt is located in the memory location (78). From the saved (50) contents of the issuing processor (12b), any parameters necessary for the handling of the system management interrupt are retrieved (54).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether Independent Claims 1, 14, and 16 are obvious under 35 USC 103(a) over the combination of Applicants' allegedly admitted prior art, *Goodman*, and *Smith.*

VII. ARGUMENT

Rejections of Independent Claims 1, 14, and 16 under 35 USC 103(a) based on Applicants' allegedly Admitted Prior Art *Goodman*, and *Smith.*

Claims 1, 4-8, 16, and 19-23 were rejected by the Examiner under 35 U.S.C. §103(a) as being obvious over Applicants' allegedly admitted prior art in view of U.S. Patent 6,282,601 issued to Goodman et al. ("*Goodman*"), and further in view of U.S. Patent 3,643,227 to Smith et al. ("*Smith*"). Appellants respectfully disagree and submit that a *prima facie* obviousness rejection has not been established. Appellants contend that the combination of

Applicants' allegedly admitted prior art, *Goodman*, and *Smith* is improper and that it does not teach or suggest all of the elements of Appellants' independent claims.

A. Failure to Establish a Prima Facie Rejection - Improper Combination with Goodman

The Final Office action mailed December 27, 2006 states at page 10:

> In response to Applicants' argument, while the Examiner agrees with Applicants that Goodman discloses dedicating the boot processor to handle the interrupt management, Applicants are reminded that the rejection is based on a combination of references. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.

Appellants would like to reiterate that, in this case, the combination of references relied upon by the Examiner is **improper due to Goodman**. References may not be combined unless there is some **motivation or suggestion** in the prior art for the combination. In particular, references *cannot be combined* if their combination would *teach away* from the claimed invention. Appellants have focused on Goodman to demonstrate that Goodman *teaches away* from the claimed invention, and thus, any combination of references involving Goodman is improper. A *prima facie* case of obviousness has not been established by the Examiner because the Examiner relied on this improper combination of references involving Goodman.

Here, the asserted prior art identified by the Examiner *cannot* properly be combined with Goodman because Goodman *teaches away* from the invention. A reference teaches away from the invention when a person of ordinary skill in the art, upon reading the reference, would be led down a path that is divergent from the path of the patent applicant. *See Tec Air, Inc. v. Denso Mfg. Mich. Inc.*, 192 F.3d 1353 (Fed. Cir. 1999); *In re Gurley*, 27 F.3d 551, 553 (Fed. Cir. 1994) (explaining that a reference teaches away if it suggests a line of development that is unlikely to produce the result sought by the applicant). It is improper to

combine references that teach away from their combination. *In re Grasselli*, 713 F.2d 731 (Fed. Cir. 1983); MPEP 2145.

Goodman teaches away from the claimed invention by *expressly* providing that all system management interrupts are to be handled by a **single, dedicated** processor. Goodman plainly discloses that "only the boot processor" is to be involved in handling system management interrupts (column 4, line 57). There is no teaching from Goodman to suggest that *any* processor other than the single boot processor is operable to handle a system management interrupt. Nowhere does Goodman suggest that multiple processors could be used to handle system management interrupts, as explicitly claimed in the present application. A plain reading of Goodman would lead a person of ordinary skill to conclude that only a **single, dedicated** processor is responsible for handling a system management interrupt, in direct contrast to the claims of the present invention.

Appellants would like to emphasize that Goodman must be considered in its **entirety**. It is well established that a prior art reference must be considered in its entirety, including those portions that point to the nonobviousness of the invention at issue. The relevant section of the MPEP, 2141.02, states that "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention." (emphasis in original). This section of the MPEP includes a detailed discussion of the W.L. Gore & Assoc., Inc. v. Garlock, Inc., 721 F.2d 1540, 1550-51 (Fed. Cir. 1983) case, and the fact that the reference discussed in that case, when read as a whole, would not suggest the claimed invention. In *In re Hedges*, 783 F.2d 1038 (Fed. Cir. 1986), the Federal Circuit plainly stated that "the prior art as a whole must be considered." Id. at 1041. "[I]t is impermissible within the framework of section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what

such reference fairly suggests to one of ordinary skill in the art." *In re Wesslau*, 353 F.2d 238, 241 (CCPA 1965).

Here, when considering the teachings of Goodman as a whole, a person of ordinary skill would be encouraged to use only a **single, dedicated** processor as the processor responsible for handling a system management interrupt. When considered as a whole, the prior art counsels *directly* against Appellants' invention. This is "strong evidence" of the nonobviousness of the invention because Goodman teaches a solution that is the *opposite* of the invention of the present application. "[M]atter in the prior art which counsels against doing what the inventor did is strong evidence that the inventor's solution is not obvious." Johnson & Son, Inc. v. Gillette Co., 1989 WL 87374, *42, Civ. A. Nos. 83-2657-N, 83-3201-N, (D. Mass. 1989). Therefore, Appellants respectfully submit that Goodman must be considered in its entirety, and, when Goodman is considered in its entirety, Goodman teaches away from the claimed invention. As a result, a rejection of the pending claims on the basis of any combination involving Goodman is improper, and for this reason alone, a *prima facie* case of obviousness has not been established.

B. Failure to Establish a Prima Facie Rejection - All Claim Limitations are not Taught or Suggested

Goodman, taken alone or in combination with Smith and Applicants' allegedly admitted prior art, does not teach or suggest all of the claim limitations of the present invention. Namely, because Goodman teaches a single, dedicated processor responsible for handling a system management interrupt, Goodman fails to teach or suggest that **each** processor of the multiple processors is operable to process a system management interrupt and **none** of the processors are dedicated to processing system management interrupts. Additionally, Applicants' allegedly admitted prior art fails to remedy this deficiency, as the cited portions of the

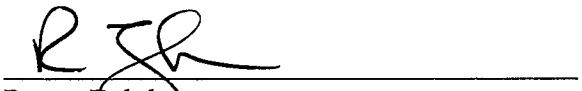
Background of Appellants' Specification discuss how, in a multi-processor system, a second processor may be *unable* to properly process a system management interrupt. (Specification, p.4, lines 18-22) Finally, Smith also fails to remedy the deficiencies of Goodman and Applicants' allegedly admitted prior art, as Smith fails to even discuss system interrupts. Thus, because the combination of references fails to teach or suggest all of the claim limitations of the present invention, a *prima facie* case of obviousness has not been established.

As a *prima facie* obviousness rejection has not been established against Appellants' independent claims, Appellants respectfully request allowance of independent claims 1, 16, and 22. Additionally, because pending claims 4-8, 19-21, and 23 depend from these independent claims, Appellants respectfully request allowance of all pending claims.

SUMMARY

Appellants respectfully request review and reversal of the obviousness rejections of Claims 1, 16, and 22 and subsequent allowance of all remaining claims.

Respectfully submitted,



Roger Fulghum
Reg. No. 39,678

Baker Botts L.L.P.
910 Louisiana
One Shell Plaza
Houston, Texas 77002-4995
(713) 229-1707

Baker Botts Docket Number: 016295.0624

Date: October 3, 2007

APPENDIX A - CLAIMS INVOLVED IN APPEAL

1. (Previously Amended) A method for handling system management interrupts in a multiprocessor computer system, wherein each processor of the multiple processors of the computer system is operable to process a system management interrupt and wherein none of the processors are dedicated to processing the system management interrupts of the computer system, comprising the steps of:

writing a predetermined signature to a predetermined register of a first processor;

executing in the first processor a command of a software application to cause the first processor to initiate a system management interrupt;

receiving at each processor an instruction that the system management interrupt has been issued;

entering system management mode at each processor;

saving register contents of each processor to a memory space associated with each respective processor;

selecting from among the multiple processors a second processor as a system management interrupt handler, the selection of the second processor being accomplished according to an arbitration scheme;

scanning the contents of the memory space associated with each processor; and

when the selected second processor locates the saved predetermined signature in one of the memory spaces associated with the processors of the computers system, using the contents of the memory space associated with the predetermined signature for any parameters necessary for the handling of the system management interrupt.

2-3. (Cancelled).

4. (Previously Amended) The method for issuing and handling system management interrupts in the multiprocessor computer system of claim 1, wherein the first processor and the second processor are the same processor.

5. (Previously Amended) The method for issuing and handling system management interrupts in the multiprocessor computer system of claim 1, wherein the step of executing in the first processor the command of the software application to cause the first processor to initiate the system management interrupt comprises the step of executing a software instruction causing the first processor to write to a predetermined port of a chip set of the computer system.

6. (Previously Amended) The method for issuing and handling system management interrupts in the multiprocessor computer system of claim 5, wherein the predetermined port of the chip set resides in a PCI bridge of the chip set.

7. (Previously Amended) The method for issuing and handling system management interrupts in the multiprocessor computer system of claim 5, wherein the predetermined port of the chip set resides in an expansion bridge of the chip set.

8. (Previously Amended) The method for issuing and handling system management interrupts in the multiprocessor computer system of claim 7, further comprising the step of issuing from the expansion bridge the instruction causing each of the processors of the system to enter system management mode.

9-15. (Cancelled).

16. (Previously Amended) A method for handling system management interrupts in a multiprocessor computer system in which each of the processors of the computer system is operable to handle software system management interrupts and wherein none of the processors are dedicated to processing the system management interrupts of the computer system, comprising the steps of:

issuing an instruction from a first processor of the system to a chip set of the computer system;

receiving the instruction at the chip set of the computer system and, in response, issuing a command causing the processors of the system to enter system management mode;

writing a software system management interrupt signature to a predetermined register of the first processor as an indication that the first processor issued the command that caused the processors of the system to enter system management mode;

writing contents of the registers of each processor to a memory location, the memory location including a memory space reserved for and associated with the register contents of each processor;

selecting a second processor, the selection of the second processor as the system management interrupt handler being accomplished according to an arbitration scheme;

transmitting a software system management interrupt to the second processor of the computer system, the second processor including a system management interrupt handler, and the second processor locating, in response to the receipt of the software system management interrupt, the software system management interrupt signature in the memory location; and

retrieving for use by the system management interrupt handler as parameters register contents saved by the first processor to the memory space associated with the software system management interrupt.

17-18. (Cancelled).

19. (Previously Amended) The method for handling system management interrupts in the multiprocessor computer system of claim 16, wherein the instruction from the first processor to the chip set of the computer system is a write command to a predetermined port of the chip set.

20. (Previously Amended) The method for handling system management interrupts in the multiprocessor computer system of claim 19, wherein the write command is received and the software system management interrupt is issued by a PCI bridge of the chip set.

21. (Previously Amended) The method for handling system management interrupts in the multiprocessor computer system of claim 19, wherein the write command is received and the software system management interrupt is issued by an expansion bus bridge of the chip set.

22. (Previously Amended) A method for handling a system management interrupt in a multiprocessor computer system, wherein each of the multiple processors of the computer system is capable of operating as a system management interrupt handler and wherein none of the processors are dedicated to processing the system management interrupt of the computer system, comprising the steps of:

receiving at each of the processors an instruction to enter a mode associated with the issuance of a system management interrupt;

selecting a designated processor from among the set of processors capable of operating as a system management interrupt handler, as a system management interrupt handler, the selection of the designated processor being accomplished according to an arbitration scheme;

scanning the memory location containing saved contents of each processor of the computer system;

locating in the memory location a signature identifying the saved contents of the processor that issued an instruction that caused the system management interrupt; and

retrieving from the saved contents of the issuing processor parameters necessary for handling of the system management interrupt.

23. (Previously Amended) The method for a handling system management interrupt in the multiprocessor system of claim 22, wherein the designated processor is not the processor that issued the instruction that caused the system management interrupt.

24-27. (Cancelled).

APPENDIX B: EVIDENCE

NONE

APPENDIX C: RELATED PROCEEDINGS

NONE